# Plane Extraction and Error Modeling of 3D data

Diego Viejo and Miguel Cazorla
Dept. de Ciencia de la Computación e I. A.
Universidad de Alicante
P.O. Box 99 03080 Alicante
{dviejo,miguel}@dccia.ua.es

***Abstract*— We are interested in using natural landmarks obtained by a stereo system not only in SLAM-like algorithms but also feature extraction, map building, and so on. Using a stereo camera we can extract planes and geometrical primitives like that. In order to use these primitives a perceptual model of landmarks is needed, due to error model can improve the results. In this paper we present a method to get the perceptual model of the plane extraction process. We will show that the use of this model improves the results and some points about its possible use are detailed.**

***Keywords*— 3D robot mapping, plane extraction, error modeling.**

## I. INTRODUCTION

Our main goal in this work is to obtain planar patches from 3D data. Knowing the perceptual error model is a very important aspect for any automatic map building approach. A nice error model allow us to deal with measured data uncertainty, and introducing it in any algorithm we can improve its accuracy in fields as feature extraction, mapping, localization, SLAM, etc. Our goal is to extract planar patches from the raw 3D data, as accurate as possible.

We have used the robot shown in Figure 1. It is equipped with a laser Sick and a Digiclops tri-stereo camera. We have calibrated both sensors in order to get them synchronized. In spite of handle raw 3D points data, we will use a geometric feature extraction approach, as we will see below, that give us plane patches belonging to the walls of an indoor 3D scene. We want to use these planes as landmarks for the SLAM algorithm. Then we show a method for quantifying the error committed in this process. The measurement error is not obtained from the raw data, but from computed planes, and so we can assume a virtual sensor, which retrieves us plane patches directly from a 3D scene, and we know its error model. The error model will be introduced in the plane extraction process and compared with the previous results.

Our final goal (not covered in this paper) is the simultaneous localization and mapping (SLAM) problem. SLAM consists of recovering a spatial map of the environment where an autonomous vehicle or robot is moving on, while it attempts to estimate its own pose (location and orientation) relative to the map. There are a lot of applications that try to resolve the problem of SLAM ([6], [2], [5], [8]).

Other approaches for 3D mapping use EM algorithm, Hough transform, or grid based ([3], [7], [9], [12], [13]). Some of them work directly with raw 3D data and some others work with an intermediate plane extraction process. However, no modeling error appears in these works. This occurs because some data come from laser, which has a low associated error. In case of stereo camera, the error is greater than laser and it has to be modeled.


Fig. 1. Our mobile robot equipped with the stereo vision system and a laser Sick.

The rest of the paper is organized as follows. In Section II we present a explanation on how plane patches are extracted from stereo data. Then, Section III explains the method used for getting the error and shows the results of such process. The error is divided in three: error in $Z$ axis, in $X$ and $Y$ axis, and in angle. Section IV shows two experiments: first, incorporating the error model into the plane extraction process; and second, showing the result of obtaining a complete 3D map from an indoor environment. We conclude in Section V with our future work.

## II. PLANE EXTRACTION

We obtain a set of 3D points from a stereo camera in each pose where the robot is (see Figure 4 left). This set of 3D points belongs to a 3D scene (see Figure 4 middle). From the set of points we want to know some information about surfaces in the environment. So, we use a surface normal vector estimation procedure that help us to decide when a 3D point belongs to a plane surface [10], [11]

It is well known that the normal of a measured point can be estimated by the eigenvector belonging to the smallest eigenvalue of the 3x3 matrix whose elements are the tree coordinate variances and the corresponding co-variances from its neighbor points. Those neighbor points are inside a cubic area centered into the point $p$ what we want to compute its normal. The size of this area is about 20 cm. The vector from the point $p$ and each point inside the neighbor area is computed and added to the co-variance matrix. Once the covariance matrix is built, we perform a eigenvalue decomposition. The resulting normal vector corresponds to the smallest eigenvalue. Thus, from eigenvalues $\sigma_1^2 \leq \sigma_2^2 \leq \sigma_3^2$ we can compute a *thickness angle* [4] that characterizes how long points rise from the plane they belong to. If points belong to a perfect plane surface, thickness angle will be a small value, but if points don't belong to a plane or it are so noisy that is no possible perform a reliable normal estimation, thickness angle will be a larger value. We can see the result of normal estimation in figure 2.

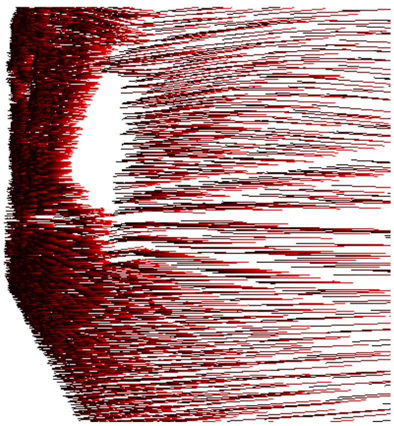

Fig. 2. Normal vector estimated from a 3D scene.

Once we have estimated normal vectors, we compute vertical planes from the scene. Firstly, we cluster neighbor points with similar normal vectors. We consider that two points have similar normal vectors when the angle between them is under a threshold, usually 20 degrees. Then, we recompute the thickness angle from the points in each cluster and eliminate those clusters with a high thickness angle in order to avoid clusters that doesn't fit with a plane (image 3). At the last step, a final fusion of similar planes is performed. Two planes are fused if they are neighbors (there's no any other plane between them), their normals are similar (idem that for a pair of points) and them are close enough. The parameters of the resulting fuse plane are computed as the mean of the parameters of the two source planes. After fuse, small planes are removed in order to retrieve the main planes of the scene. Bounds of the remaining planes are computed by projecting its belonging points on the plane. Finally, plane texture is computed from the reference image by transforming each piece on the plane surface (we take 1 $cm^2$ piece size) into a pixel on the reference image, and getting its grey level. A pair of examples of the result of this process can be seen in figure 4-right.

This plane extraction process give us some advantages in front of deal with raw data. Firstly, we do an important reduction of the 3D scene complexity, from about 300,000 3D points to a few number of planes (usually less than 10). Thus, the vector normal estimation process improves the

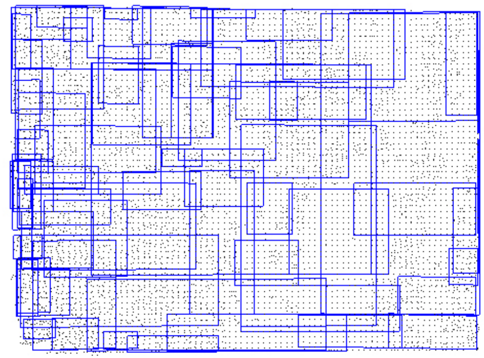

Fig. 3. Planes extracted from a 3D scene before fuse step.

accuracy of the resulting model int two ways, in one hand, the thickness angle computed from the singular values makes possible discard points from very noisy areas, and in the other hand, the singular value decomposition performed over the points itself reduces noise.

## III. ERROR MODELLING

We are evaluating the error done in the plane extraction process. For this task, we divide the error in three different ones: error in the $Z$ axis; error in the $X$ and $Y$ axis; and angular error. Also, we are interested in checking how different textures could affect to this process. For this work we are selected the two well differentiated textures shown at Figure 5, as these two textures are the ones predominant in our environment.

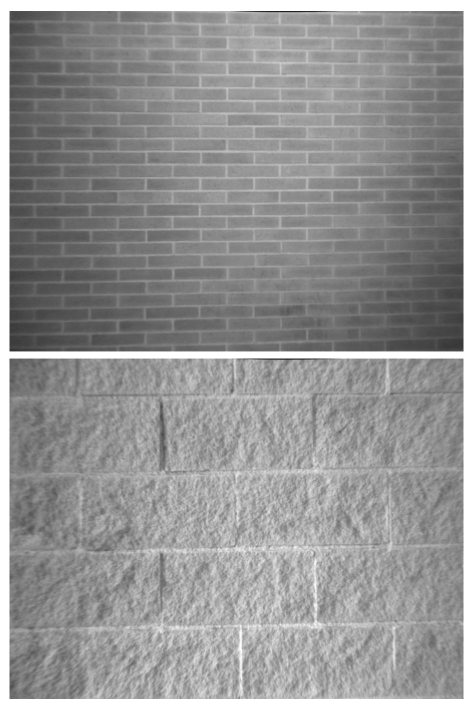

Fig. 5. Textures used for the experiments. At the top, the texture marked as *Texture1* and at the bottom the texture marked as *Texture2*.

As we explained before, in our experiments we are going to use a Magellan Pro mobile robot equipped with a Digiclops

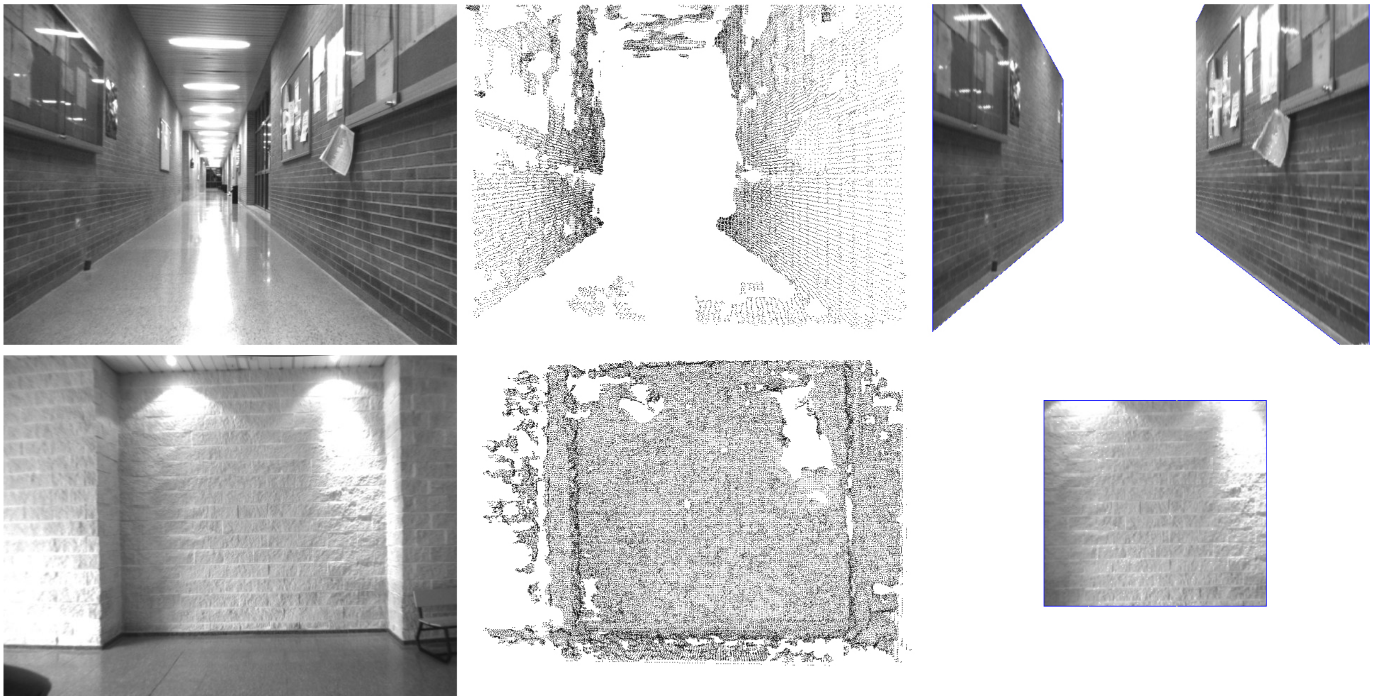

Fig. 4. Plane extraction process. In the left column, real world pictures. In the middle column, sets of 3D points from the stereo camera. In the right column, resulting planes from our plane extraction algorithm.

stereo camera from Point Grey and a LMS-200 SICK range laser. Laser measurements are more accurate ($\leq 1mm$) than stereo camera, so we are going to use laser data as ground truth to compare with the results of the plane extraction process. First, we have to find the transformation between stereo camera and laser coordinate systems. For this task, we have checked that planes placed near the stereo camera (about 0.5 meters or less) have an error of 1mm or less (the laser error). So, we place the robot at an approximated distance of 0.5 meters and take data from the stereo camera and from laser. Stereo data is processed in order to obtain the planes. Laser data is processed using the Recursive Iterative End Point Fit Algorithm [1], which provides segments from the laser data. In the calibration process, the largest segment is used for obtaining the ground truth, i.e., the relative position of the laser from the camera.

### *A. Calculating the error in* Z *axis*

In this section, the error in the *Z* axis is calculated. *Z* axis is located perpendicular to the image plane and it is negative for points in front of the image plane. The camera is situated from 1 to 5 meters, at intervals of half a meter. Although the camera has a range of more than 8 meters, distances above 5 meters have a problem: textures begins to disappear (the camera is not able to recover 3D points). The plane extraction process tries to extract the bigger plane and it is used as reference.

Figure 7 (left) shows the measured error (mean and variance). As we can observe, the mean is almost the same but the variance is significantly greater in the first texture (bricks). This can be due to that sometimes the plane extraction process is not able to get 3D data and thus it extracts a bad plane or even does not extract any plane. We can conclude that these two textures have almost the same error and that this error is always negative, i.e., the estimated planes are further away than the real ones. Furthermore, the variance of this error is produced (mainly) by the absence of 3D data. This must be considered when using this model.

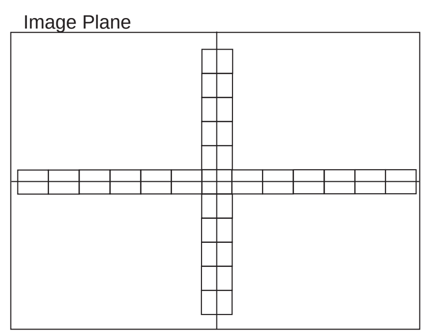


Fig. 8. Squares used for the error in *X* and *Y* axis.

### *B. Calculating the error in* X *and* Y *axis*

Additional errors are calculated: error in *X* axis (positive at the right of the image center) and in *Y* axis (positive above of the image center). We want to know if error changes as planes gets further away the center of the image, for a constant distance. To do that, we have fixed the robot to a certain distance of the wall and we have taken several 3D data. For each 3D scene, we have divided it in sets of points, each of them formed by points to a certain distance of the image center along *X* and *Y* axis. Figure 8 shows the areas in which we divide the scene. The plane extraction process is only applied over the points

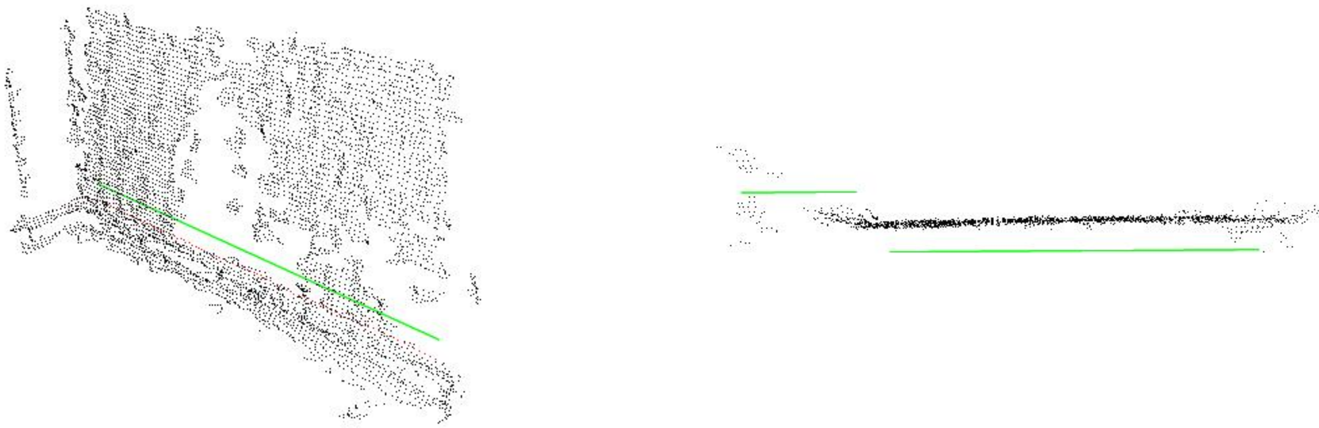

Fig. 6. Laser (continuos line) and stereo data overlapped. We use the difference between laser data and 3D data in order to get calibrated both sensors.

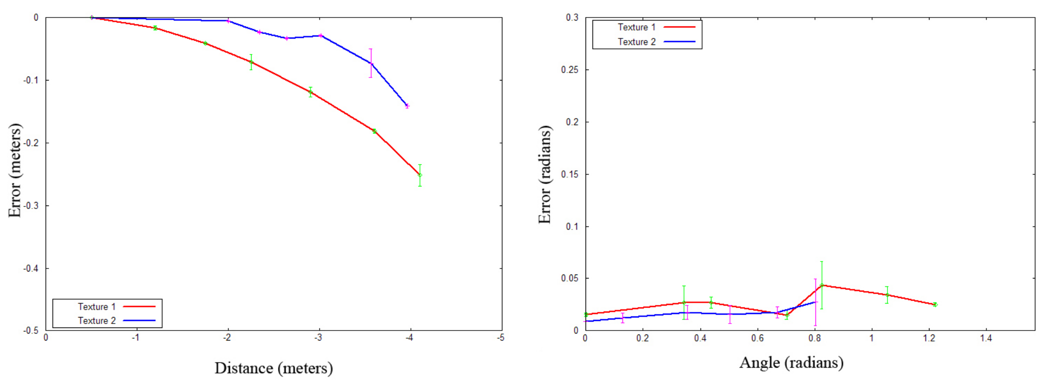


Fig. 7. Error in the $Z$ axis (left) and with respect of angle (right).

in each set in spite of apply it over all the points of the scene. As result, we obtain a plane for each subset of points and we compute its error as is described in the previous section.

As shown in Figure 9, this error is not significative. The error remains moreover constant as planes gets further away both in $X$ and $Y$ axis.

### C. Calculating the angular error

Figure 7 shows the results of angular error. Here, we use the angle of the segment extracted from laser data and it is compared with the plane angle. The error is always below 0.05 radians (3 degrees) and it has a very low variance. This is a good thing for us, so we can trust in angle.

## IV. Experiments

In this section we present two experiments done incorporating the error model.

### A. Incorporating the error model into the plane extraction process

Here, we have incorporated the previous error model into the plane extraction process. We have compared the error without and with the error model. Figure 10 shows a comparison between the error with and without the error model. As we can see, the error has decreased significantly. We have selected the $Z$ axis error, because it is the greatest one.

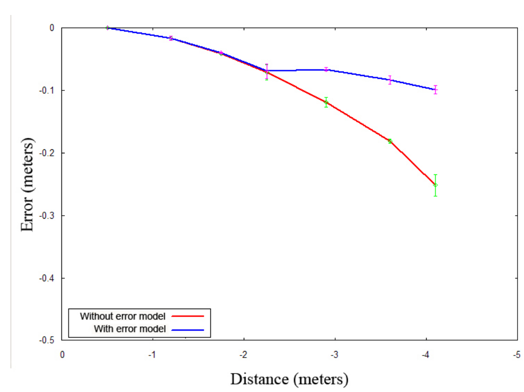


Fig. 10. Comparing the error with and without error model.

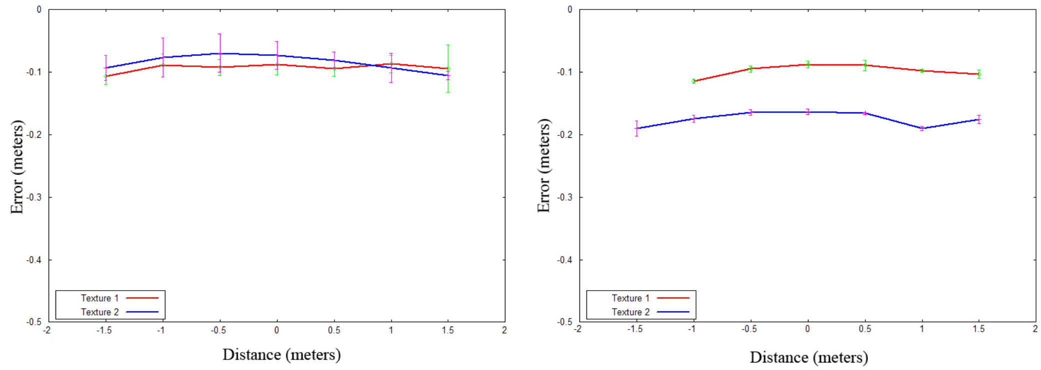


Fig. 9. Error in the $X$ (left) and $Y$ (right) axis.

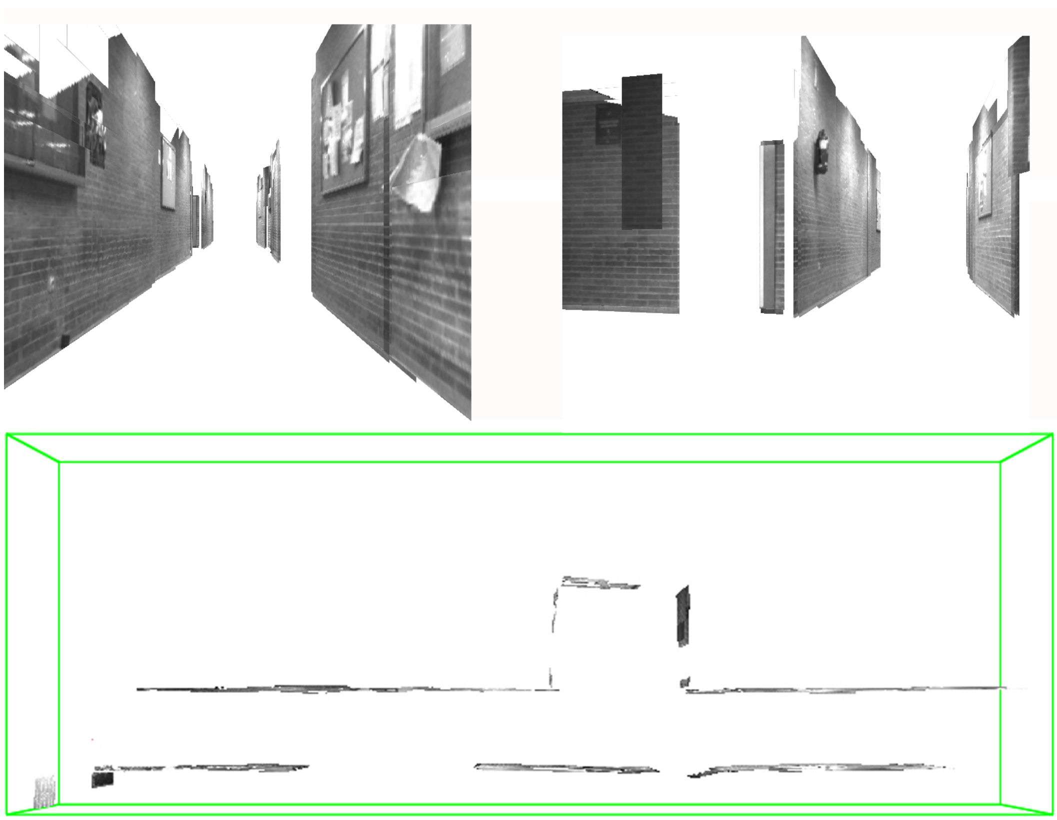

Fig. 11. Reconstruction of an indoor environment (below) and some details of the reconstruction.

### *B. Map building*

As a final result, we show in Figure 11 a complete reconstruction of an indoor environment.

## V. Conclusions and future work

In this paper we have studied the error in the process of plane extraction. This error is essential in SLAM problems, where a perceptual model is needed. We have obtained some conclusions: data at a distance above 4 or 5 meters are too noisy; planes are always obtained further from the true one; errors in the $X$ and $Y$ axis is very similar and there is no significant variation in the error; the angle of a calculated plane is good enough for SLAM-like algorithms. We have also found that the lack of texture is an important problem, due to the camera is unable to obtain some textures (the brick case is one of this) at a certain distance. In a previous work [11] we tried to solve this problem projecting a pattern over the environment, but results have to be improved.

There is an immediate application of this knowledge in the plane extraction process itself. We introduce the obtained error model in the fuse step in order to improve the computing of the resulting parameters of the plane fused.

As future work, we are planning to continue comparing the error obtained in different textures, in order to get more information about this error. We plan also to find a relation between texture and distance (to which distance a texture is good) and to move to outdoor environments.

Besides, our main interest is using this error model for solving the SLAM problem. So our next step is using it as a part of a SLAM algorithm, experimenting the possibilities of planes as artificial landmarks.

## ACKNOWLEDGMENT

This work has been supported by project GV06/134 from Generalitat Valenciana (Spain) and project DPI2005-01280 from Ministerio de Educación y Ciencia (Spain).

## REFERENCES

[1] R. Duda and R. Hart. "Classification and Scene Analysis". John Wiley and Sons. 1973

[2] G. Dissanayake, P. Newman, S. Clark, H. Durrant-White, M. Csorba. *A solution to the simultaneus localization and map building (SLAM) problem*. IEEE Transactions on Robotics and Autonomation. 17 (3). pp. 229-241. 2001

[3] R. Lakaemper and L.J. Latecki. "Extended EM for Planer Approximation of 3D Data". Proceedings of the 2006 IEEE International Conference on Robotics and Automation. Orlando, Florida. May 2006.

[4] M. Martn Nevado, J. Gmez Garcia-Bermejo, E. Zalama Casanova. *Obtaining 3D models of indoor environments with a mobile robot by estimating local surface directions*. Robotics and Autonomous Systems. Elsevier.vol 48/2-3. pp. 131-143. 2004.

[5] M. Montemerlo, S. Thrun, D. Koller, B. Webgreit. *FastSLAM 2.0: An improved particle filtering algorithm for simultaneus localization and mappling that probably converges*. In proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, 2003.

[6] J.M. Porta, B.J.A. Kröse. *Appearance-based concurrent map building and localization*. Robotics and Autonomous Systems. Elsevier. vol 54. pp 159-164. 2006

[7] S. Thrun, C. Martin, Y. Liu, D. Hhnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. "A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots." IEEE Transactions on Robotics and Automation, 2003.

[8] S. Thrun and M. Montemerlo and D. Koller and B. Wegbreit and J. Nieto and E. Nebot, *FastSLAM: An Efficient Solution to the Simultaneous Localization And Mapping Problem with Unknown Data Association*, Journal of Machine Learning Research Addison-Wesley, 2004 (To appear).

[9] R. Triebel, W. Burgard and F. Dellaert , "Using Hierarchical EM to Extract Planes from 3D Range Scans". In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2005

[10] Diego Viejo, Miguel A. Cazorla "Unconstrained 3D-Mesh Generation Applied to Map Building" 9th Iberoamerican Congress on Pattern Recognition, Puebla (Mexico), Octubre, 2004.

[11] Diego Viejo, Juan Manuel Sez, Miguel A. Cazorla, Francisco Escolano "Active Stereo Based Compact Mapping" IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta (Canada), Agosto, 2005

[12] Weingarten, J. and Siegwart, R. "EKF-based 3D SLAM for Structured Environment Reconstruction". In Proceedings of IROS, Edmonton, Canada, August 2-6, 2005. (IROS'2005)

[13] Denis F. Wolf, Andrew Howard, and Gaurav S. Sukhatme, "Towards Geometric 3D Mapping of Outdoor Environments Using Mobile Robots", In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1258-1263, Aug 2005